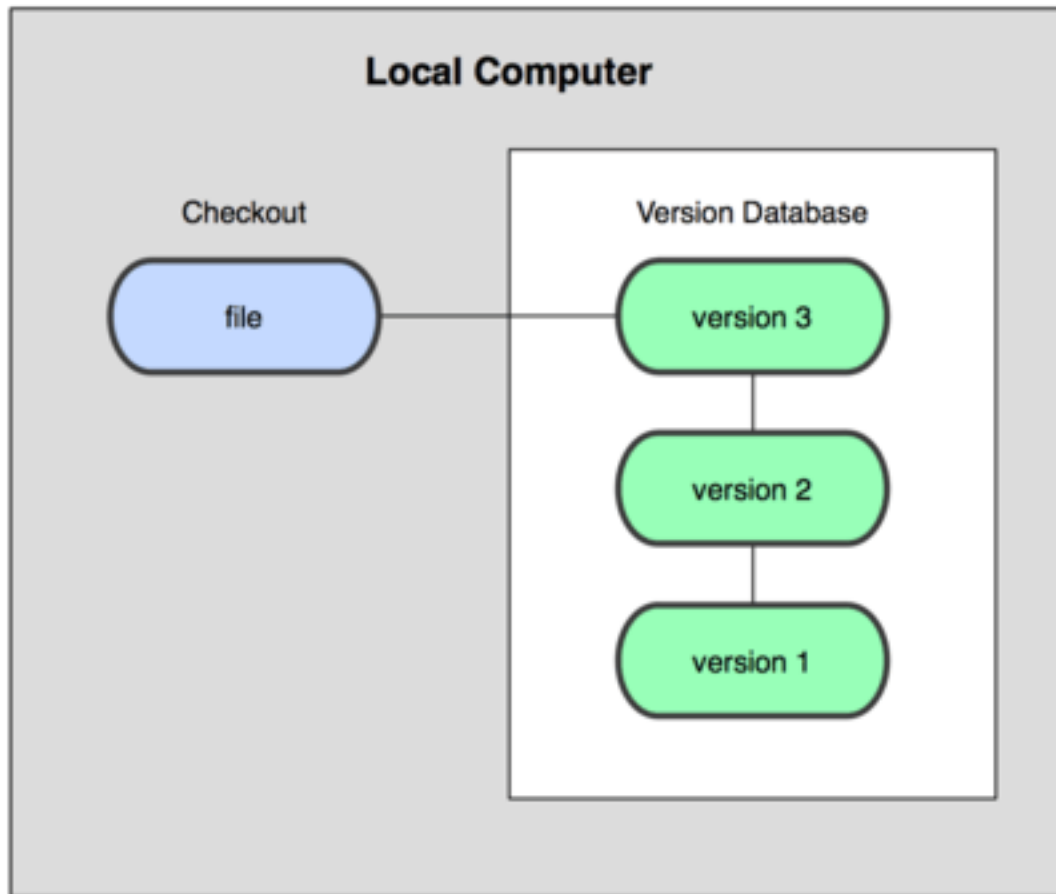


Системы контроля версий файлов

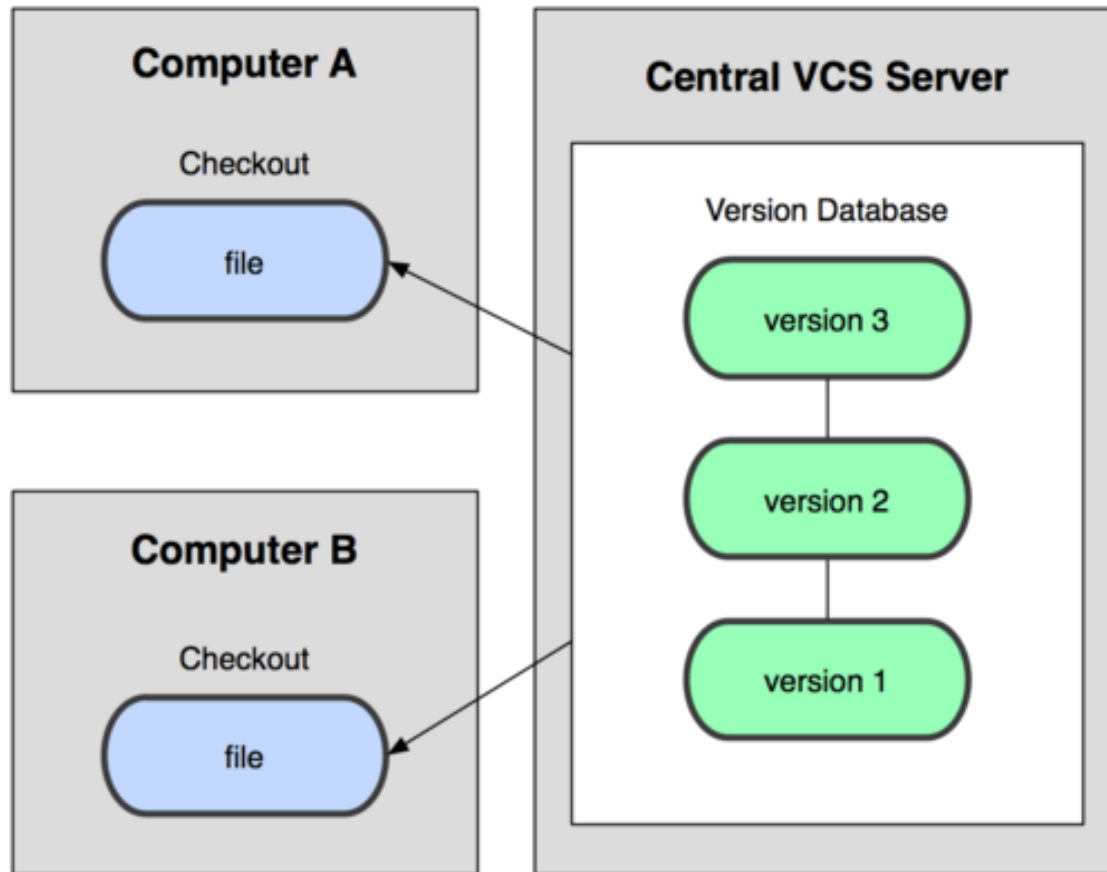
Рассматриваемые вопросы

1. Системы контроля версий
2. Основы GIT
3. Ветвления GIT

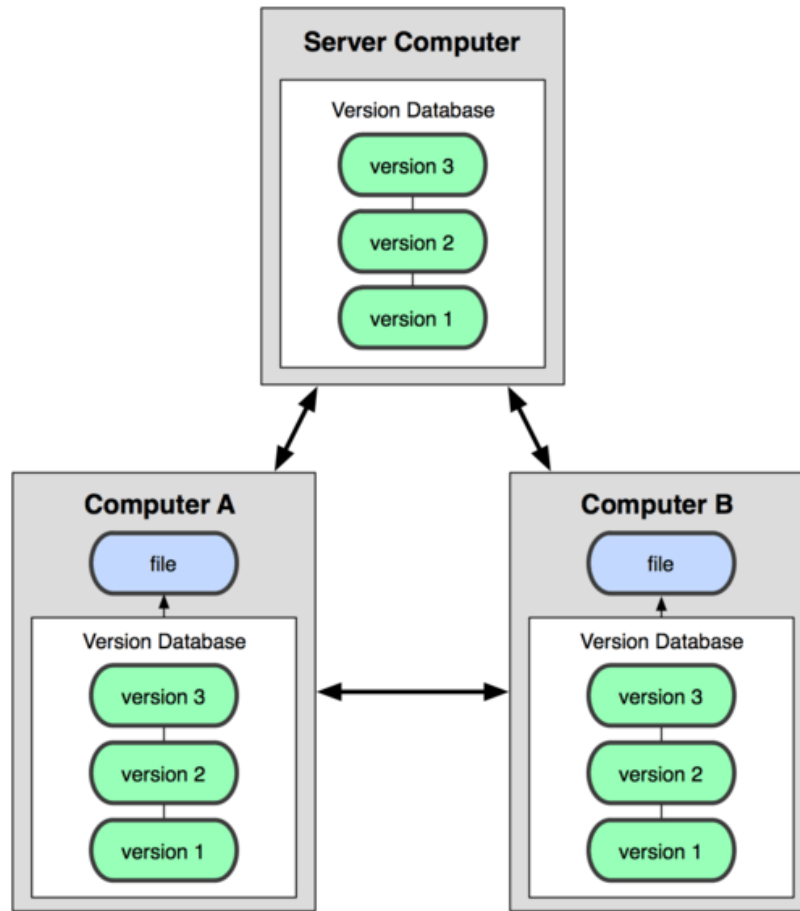
Локальные системы контроля версий



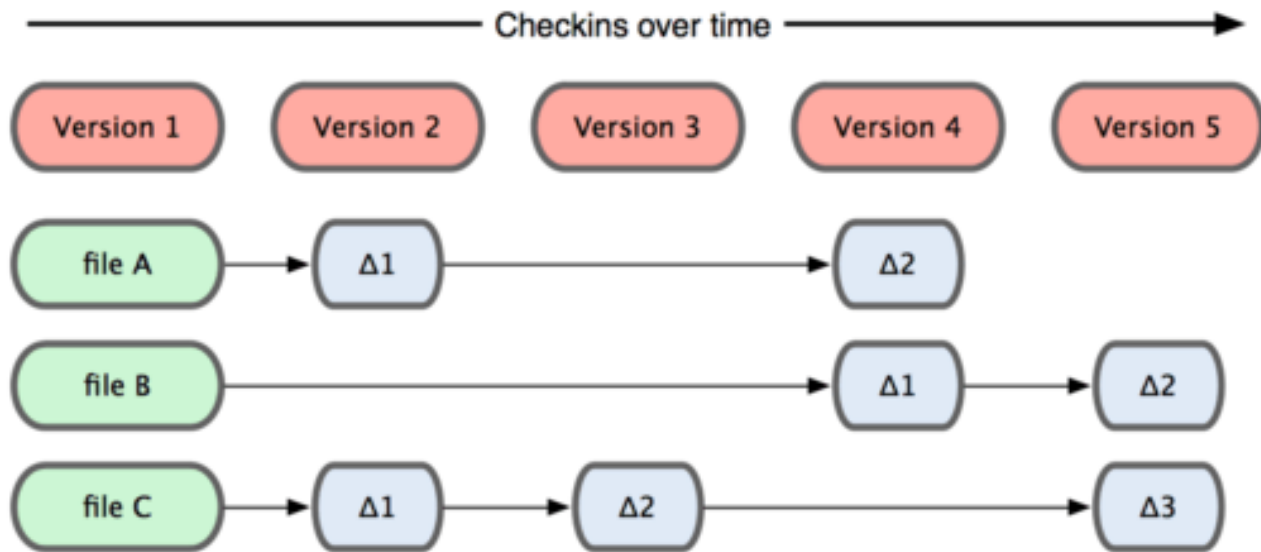
Централизованные системы контроля версий



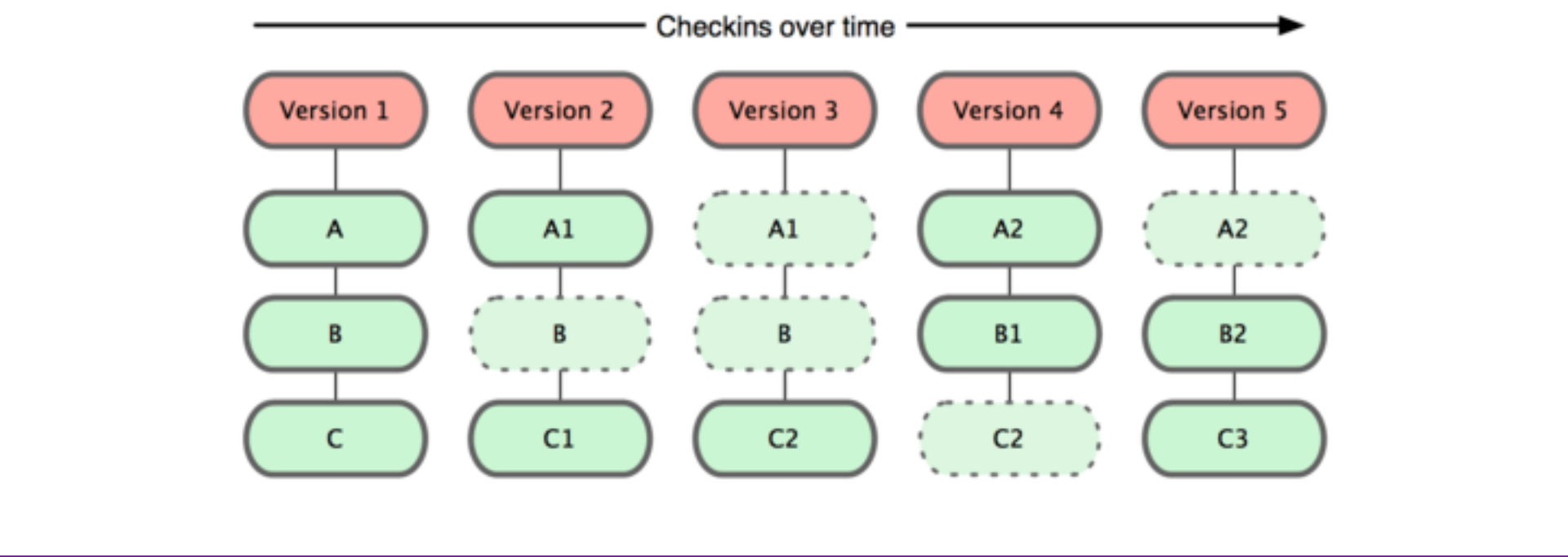
Распределённые системы контроля версий



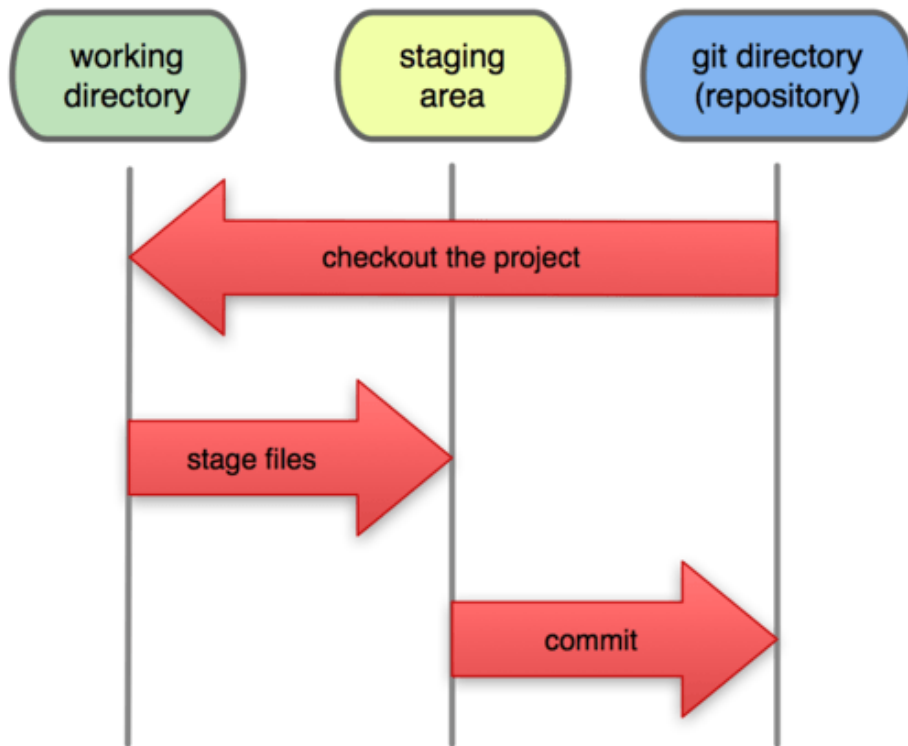
Пример хранения файлов в SVN



Пример хранения файлов в GIT



Local Operations



Создание репозитория в существующем каталоге

```
$ git init
```

```
$ git add *.c
```

```
$ git add README
```

```
$ git commit -m 'initial project version'
```

Клонирование существующего репозитория

```
$ git clone git://github.com/schacon/grit.git
```

Клонирование в конкретную директорию

```
$ git clone git://github.com/schacon/grit.git mygrit
```

Определение состояния файлов

```
$ git status
```

Фиксация изменений

```
$ git commit
```

Удаление файлов

```
$ git rm
```

Добавление удалённых репозиториев

```
$ git remote
```

```
origin
```

```
$ git remote add pb
```

```
git://github.com/paulboone/ticgit.git
```

```
$ git remote -v
```

```
origin  git://github.com/schacon/ticgit.git
```

```
pb      git://github.com/paulboone/ticgit.git
```

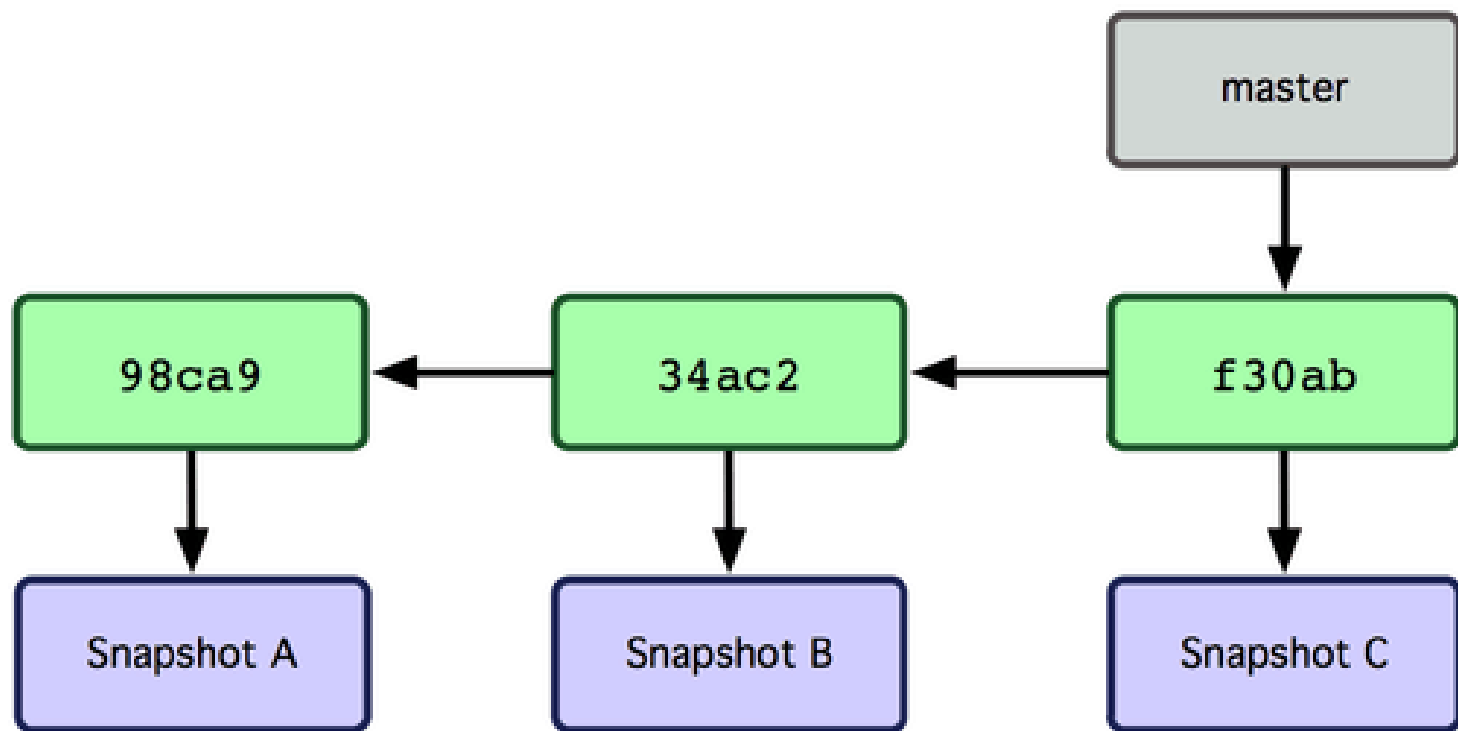
Fetch и Pull

```
$ git fetch [имя удал. сервера]
```

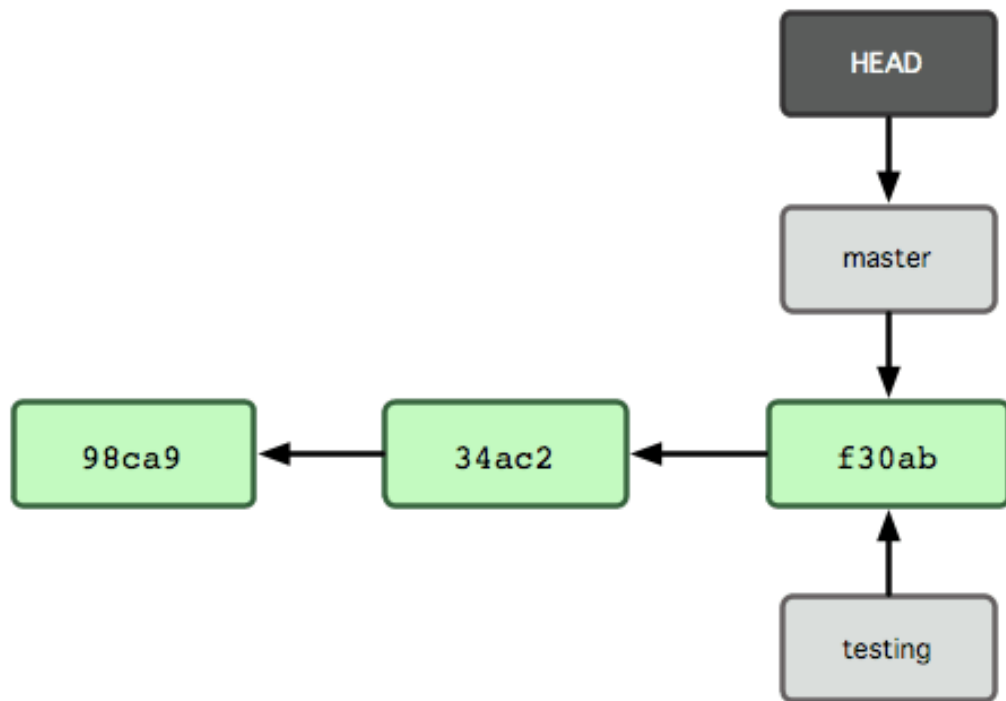
Push

```
$ git push origin master
```


Ветвления в GIT

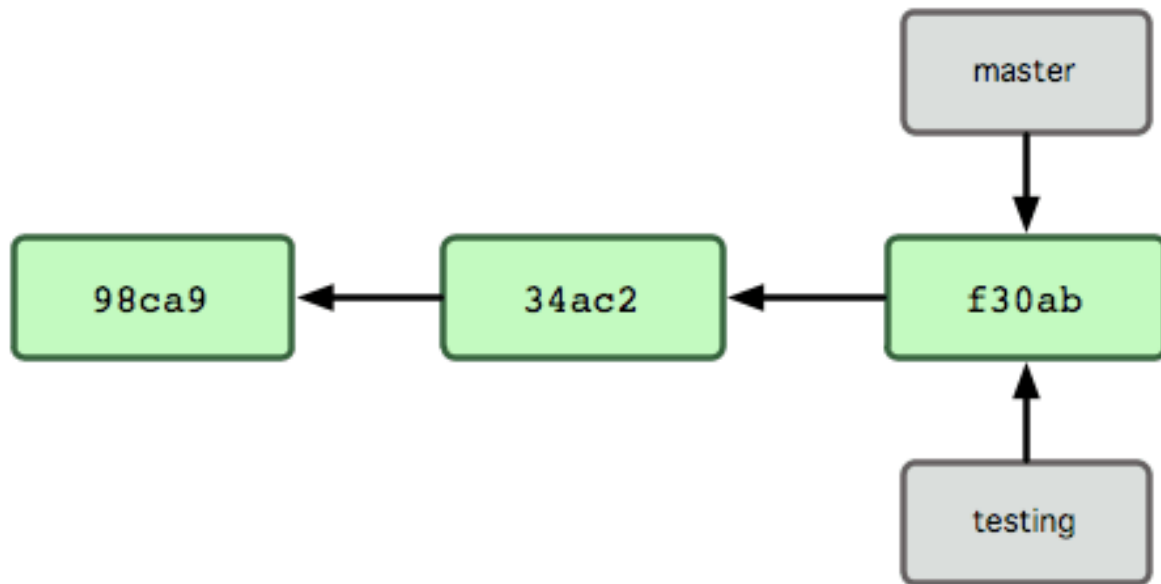


Ветвления в GIT



Ветвления в GIT

```
$ git branch testing
```



Слияние веток

```
$ git merge branchname
```

Вопросы для самоконтроля

1. Назовите основные команды GIT.
2. Как реализована работа с ветками в GIT?
3. Для чего применяется версионность в программах?